**Weekly Report – W5 Spring 2023**

Task & Problem

1. Think about how to put the dynamics of SRA into the controller in Arduino;
2. Some other arrangements about the hardware and coding work.

Solution

1. Import dynamics of SRA into controller

The biggest challenge of this task is currently all the simulations are done via MATLAB, so all the codes are in MATLAB language rather than C/C++, on the contrary, the codes written into Arduino are all C based; to convert MATLAB language into C is a must, generally we have two ways to achieve this goal, one is to write our own code with C, the limitation is that I’m not that familiar with C like with MATLAB, there could be a lot of problems when doing such coding work, meanwhile the correctness of dynamic process hasn’t been fully verified, there might be some special cases like singularity. Another way is to sufficiently use the TMTDyn package, in which there is a function that can convert all the EOMs into mex file, which is actually in C/C++, after all the package is relatively mature, and its simulation results are reliable, if we can successfully open the mex file and confirm the I/Os, things will be much easier. The update about each approach is shown as follows.

(1). Derive new dynamics

The risk is we cannot guarantee it is correct, and to build up the EOMs for the system, we also need to know the stiffness and damping coefficients of the soft arms, for stretching in one direction, it can be relatively easy to measure the estimated values for these parameters, but for more complex motions in the 3D space, it will become unrealistic to measure, so this plan was cancelled.

(2). Using mex file in MATLAB

“mex” is the abbreviation for “MATLAB executive function”, the codes in MATLAB will be executed in sequence so that the efficiency of any for loop will not be so high, sometimes we can convert MATLAB functions into C/C++ by mex command; or we can directly write a function in C/C++ based IDE then package it into mex file. The problem is that some mex files cannot be opened even with technical software, most of them are binary, only few of them can be interpreted by C/C++, however, the variables need to be redefined after interpretation. Currently this method is beyond my ability, I will keep looking in the future.

(3). Pick out the dynamics in TMTDyn package and convert it into C/C++

This is what I am currently working on, the specific work is shown as follows,

* Redefine all the variables in a struct in C/C++;
* Solve ODE with C/C++, because using Symbolic Toolbox in MATLAB, the matrices in dynamic process are not represented by symbolic variables anymore, they are replaced by specific states at certain simulation time step (solved by ODE solvers in MATLAB, and I’m not sure if there exist certain functions could solve ODEs in C)

1. Other updates about hardware and coding work

(1). After talking with Manu, he suggested me mount the IMUs on the end effector of SRAs, and given the situation that the time is not enough, it’s not realistic to implant all the codes to C/C++ in a short time, since the data can be read in MATLAB, temporarily we can deal with all the data in MATLAB and send back all the results into Arduino to realize our final control. So in the next week I will focus on the communication between MATLAB and Arduino.

(2). Charanjit is a very talent student after have a chat with him, he has a lot of experience on programming with C and Python, and also has a good knowledge of Arduino, I think he can help me a lot about coding work. I have sent some background knowledge about our project (not including the proposal) to him to let him get a rough idea about it, currently I’m also working on a specific coding work plan, I will send it to him no later than Tuesday.